1. **What is meant by WebService?**

   A service over the web is called as WebService.

2. **Explain the history of Web Services?**

   Let's take an example of School application.

   - ✓ **School A** implemented one web application to store the student address details.
   - ✓ Now **School B** want the same kind of approach you have use for student address details. **School B** requested address module code to put in their end.
   - ✓ Now **School A** will extract the Address module code as Jar and hand over it to the **School B.**
   - ✓ In future **School A** did few more changes to the Address module again **School A** will extract the Address module code as Jar and hand over it to the **School B**.

   To overcome the above issues WebService come into picture. Web services will do the following things,

   Web service say deploy your application in server and give the service details to the consumer. I will help them to get your details.

   If you are writing any web service we should register it in UDDI to make it accessible for others.

   Let's take School A code written in Java and School B code written in .net. In this we must require a common language understand by any framework (Java/.Net/c/c++). To solve this common language XML came into the picture.

   To carry the xml data to consumer from producer we need a protocol that is SOAP was introduced.

   To convert the JAVA/.NET/C++/C specific code to SOAP we need an interface. Here SEI (Service endpoint Interface) came and it will work specific to source language and convert it to SOAP.

3. **Explain about UDDI (Universal Description, Discovery and Integration)?**

   Universal Description, Discovery and Integration is the place will register our web service to make it accessible for required consumers via Soap protocol.

4. **Explain about SEI (Service End Point Interface)?**
   It provide the access to consume the methods added a part of the wsdl.

   To convert the JAVA/.NET/C++/C specific code to SOAP we need an interface. Here SEI (Service endpoint Interface) came and it will work specific to source language and convert it to SOAP.

5. **What are all the advantages of Web Services?**

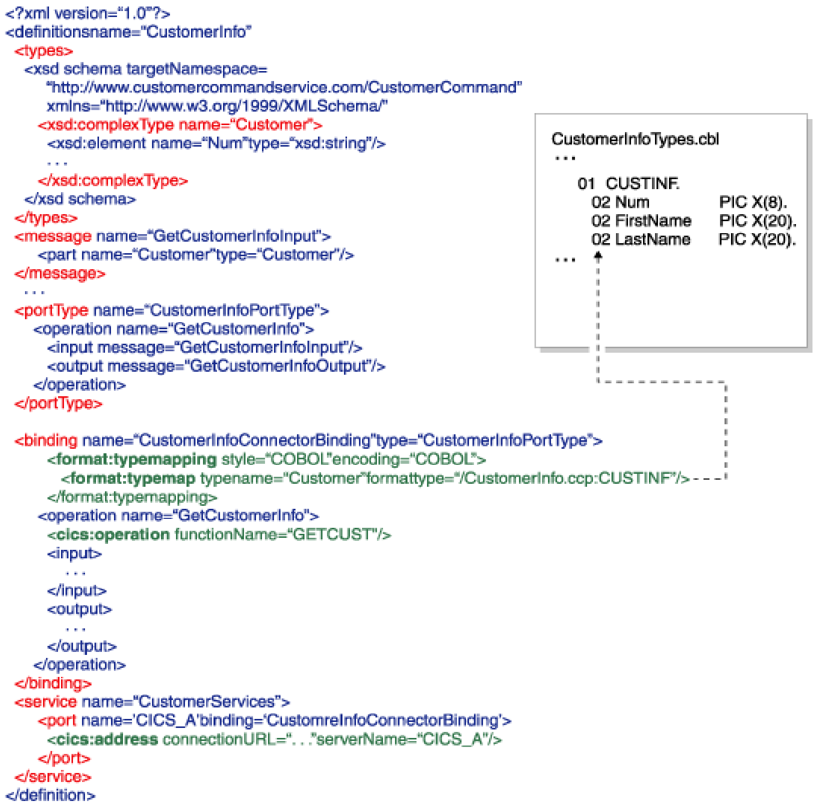
   - ✓ Interoperability - Web Services also let developers use their preferred programming languages.
   - ✓ Exposing the Existing Function on the network
   - ✓ Reusability
   - ✓ Standardized Protocol
   - ✓ Low Cost Communication

6. **Explain the WSDL (Web Service Description Language)?**

   - ✓ WSDL is an acronym for Web Services Description Language.
   - ✓ WSDL is xml document containing information about web services such as method name, method parameter and how to access it.
   - ✓ WSDL is a part of UDDI. It acts as interface between web service applications.

7. **Explain the Elements of WSDL?**

definitions: It is the root element of all WSDL documents. It defines the name of the web service, declares multiple namespaces used throughout the remainder of the document, and contains all the service elements described here.
types: its specifies the data types we are going to use in messages it may be a primitive or custom types.
messages: Defines the data elements for each operation.
Operation: It is the abstract definition of the operation for a message, such as naming a method, message queue, or business process, that will accept and process the message.
portType: Describes the operations that can be performed and the messages involved.
binding: It is the concrete protocol and data formats for the operations and messages defined for a particular port type.
service: It is a collection of related end-points encompassing the service definitions in the file; the services map the binding to the port and include any extensibility definitions.



```
<?xml version="1.0"?>
<definitionsname="CustomerInfo"
 <types>
  <xsd schema targetNamespace=
     "http://www.customercommandservice.com/CustomerCommand"
     xmlns="http://www.w3.org/1999/XMLSchema/"
    <xsd:complexType name="Customer">
     <xsd:element name="Num"type="xsd:string"/>
     . . .
    </xsd:complexType>
  </xsd schema>
 </types>
 <message name="GetCustomerInfoInput">
    <part name="Customer"type="Customer"/>
 </message>
  . . .
 <portType name="CustomerInfoPortType">
   <operation name="GetCustomerInfo">
     <input message="GetCustomerInfoInput"/>
     <output message="GetCustomerInfoOutput"/>
   </operation>
 </portType>

 <binding name="CustomerInfoConnectorBinding"type="CustomerInfoPortType">
     <format:typemapping style="COBOL"encoding="COBOL">
       <format:typemap typename="Customer"formattype="/CustomerInfo.ccp:CUSTINF"/>
     </format:typemapping>
    <operation name="GetCustomerInfo">
     <cics:operation functionName="GETCUST"/>
     <input>
       . . .
     </input>
     <output>
       . . .
     </output>
   </operation>
 </binding>
 <service name="CustomerServices">
    <port name='CICS_A'binding='CustomreInfoConnectorBinding'>
     <cics:address connectionURL=". . ."serverName="CICS_A"/>
    </port>
 </service>
</definition>
```

```
CustomerInfoTypes.cbl
...
   01  CUSTINF.
     02 Num           PIC X(8).
     02 FirstName     PIC X(20).
     02 LastName      PIC X(20).
...
```

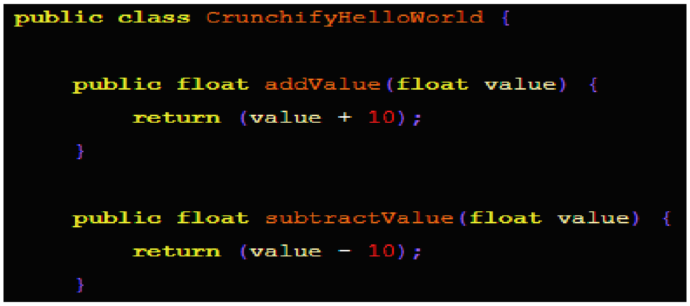
8. How to generate the stubs from WSDL using command prompt?

wsimport -keep -s src https://www.google.com?WSDL

keep -> will get the .java files
s -> To save the generated java files in specified location src

9. How to generate the web service using IDE (STS/Eclipse/RAD)?

- ✓ Create a Dynamic Web Project
- ✓ Crate the new class under package. Add methods to it.

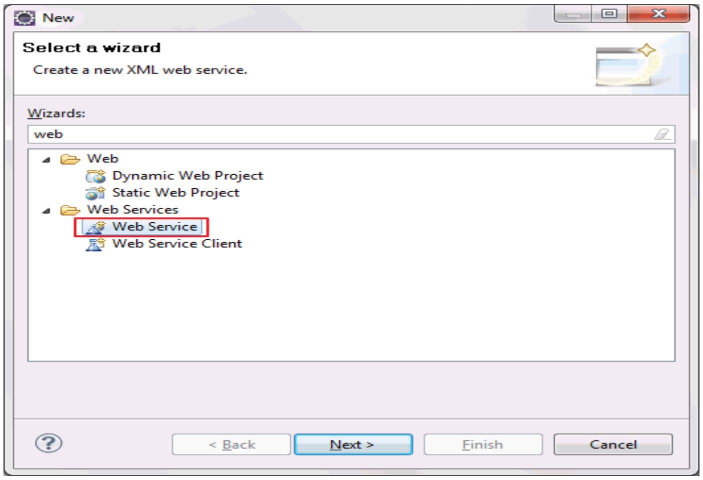


```
public class CrunchifyHelloWorld {

    public float addValue(float value) {
        return (value + 10);
    }

    public float subtractValue(float value) {
        return (value - 10);
    }
```

- ✓ Right Click on file CrunchifyHelloWorld.java -> Web Services -> Create Web Service



New

Select a wizard

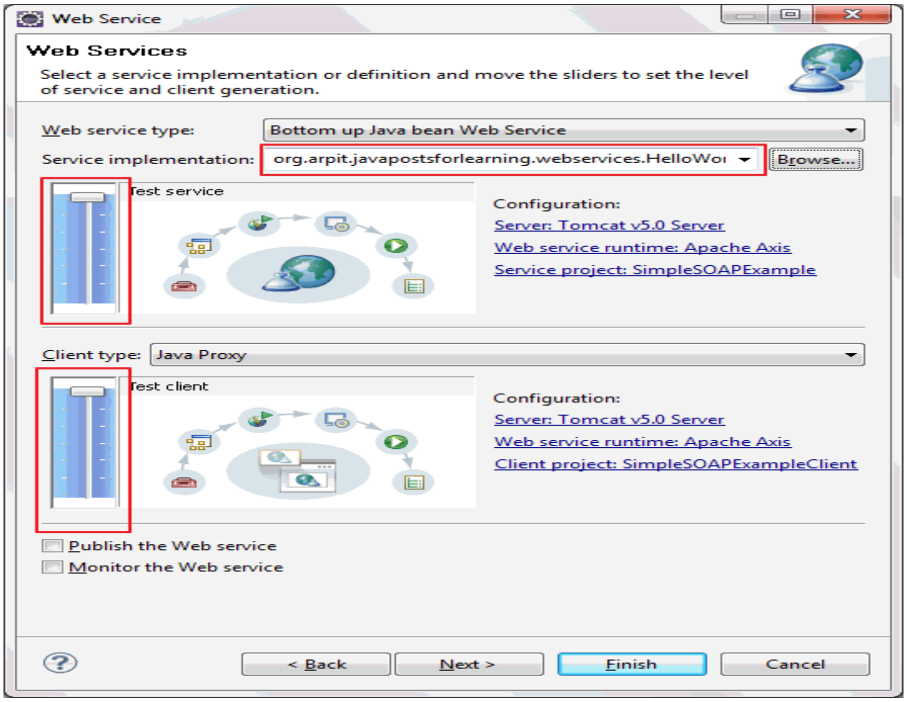
Create a new XML web service.

Wizards:

web

- Web
  - Dynamic Web Project
  - Static Web Project
- Web Services
  - Web Service
  - Web Service Client

✓ Click on next



Web Service

Web Services

Select a service implementation or definition and move the sliders to set the level of service and client generation.

Web service type: Bottom up Java bean Web Service

Service implementation: org.arpit.javapostsforlearning.webservices.HelloWoı

Test service

Configuration:
Server: Tomcat v5.0 Server
Web service runtime: Apache Axis
Service project: SimpleSOAPExample

Client type: Java Proxy

Test client

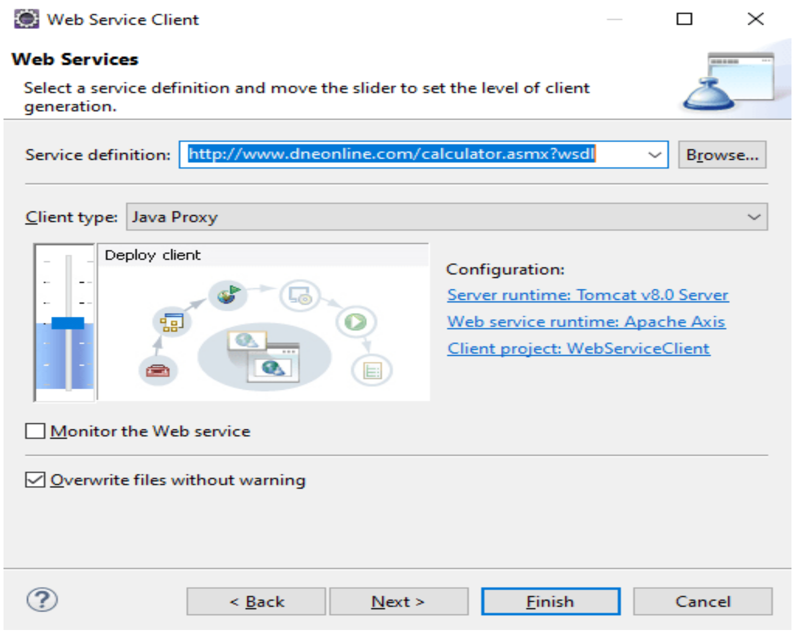
Configuration:
Server: Tomcat v5.0 Server
Web service runtime: Apache Axis
Client project: SimpleSOAPExampleClient

□ Publish the Web service
□ Monitor the Web service

✓ Click on Finish.

10. How to generate the client from WSDL?

Right click on wsdl and generate the client.



Web Service Client

Web Services

Select a service definition and move the slider to set the level of client generation.

Service definition: http://www.dneonline.com/calculator.asmx?wsdl

Client type: Java Proxy

Deploy client

Configuration:
Server runtime: Tomcat v8.0 Server
Web service runtime: Apache Axis
Client project: WebServiceClient

□ Monitor the Web service

☑ Overwrite files without warning

11. How to generate the stubs from IDE?

Add the Apache CXF dependency and add the below tasks in build.gradle file.

wsdl2java HelloWorld.wsdl

12. How to generate the WSDL from XSD?

Add the Apache CXF dependency and add the below tasks in build.gradle file.

xsd2wsdl xsdurl

13. Explain the ways to create the WebService?

Top down Approach: In top-down approach, first you design the implementation of the Web service by creating a WSDL file. You can then create the Web service skeleton Java classes from the wsdl, and add the required code.

Bottom up Approach: In Bottom up Approach first you write the java classes for the web service and then create the WSDL file and publish the web service.

Although bottom-up Web service development may be faster and easier, the top-down approach is the recommended way of creating a Web service.

14. Explain the best way to create the WebService?

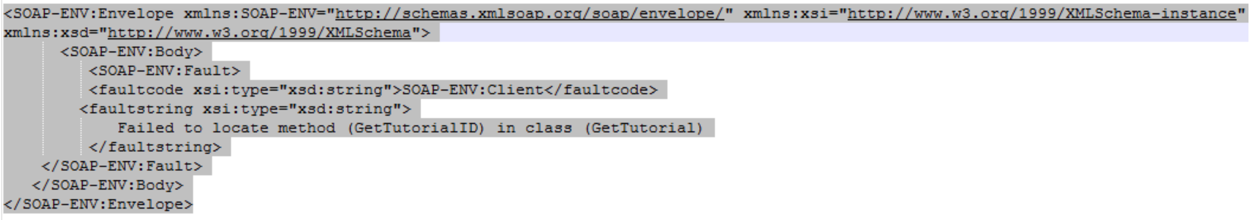
Always write the Interface and implementation class. As we put the @WebService annotation on the interface and WSDL generated from the interface if we share it will many consumers web service api's will be common for all. We can change the logics in Implementations classes it won't affect the consumers.

Always good to write the WSDL first then generate the code.

15. What is meant by SOAP?

- ✓ SOAP is an acronym for Simple Object Access Protocol.
- ✓ SOAP is a XML-based protocol for accessing web services.
- ✓ SOAP is a W3C recommendation for communication between applications.
- ✓ SOAP is XML based, so it is platform independent and language independent. In other words, it can be used with Java, .Net or PHP language on any platform.

16. Explain about soap elements?



```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
      <SOAP-ENV:Body>
         <SOAP-ENV:Fault>
         <faultcode xsi:type="xsd:string">SOAP-ENV:Client</faultcode>
        <faultstring xsi:type="xsd:string">
            Failed to locate method (GetTutorialID) in class (GetTutorial)
         </faultstring>
    </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Envelope: An Envelope element that identifies the XML document as a SOAP message – This is the containing part of the SOAP message and is used to encapsulate all the details in the SOAP message. This is the root element in the SOAP message.
Header: A Header element that contains header information – The header element can contain information such as authentication credentials which can be used by the calling application.
Body: A Body element that contains call and response information – This element is what contains the actual data which needs to be sent between the web service and the calling application.
Fault (optional): When a request is made to a SOAP web service, the response returned can be of either 2 forms which are a successful response or an error response. When a success is generated, the response from the server will always be a SOAP message. But if SOAP faults are generated, they are returned as "HTTP 500" errors.

17. Explain advantages and disadvantages of SOAP?

Advantages of Soap Web Services:

- ✓ **WS Security:** SOAP defines its own security known as WS Security.
- ✓ **Language and Platform independent:** SOAP web services can be written in any programming language and executed in any platform.

Disadvantages of Soap Web Services:

- ✓ **Slow:** SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.
- ✓ **WSDL dependent:** SOAP uses WSDL and doesn't have any other mechanism to discover the service.
- ✓ Bigger learning curve

18. How to add new method in existing service?

If you are using the "contract first" approach, the best way to add a method is indeed to modify the WSDL and then regenerate your classes with wsdl2java.

19. Explain few Web Services annotations u have used?

@Webservice: annotation makes this interface/class a web service interface. The implementing class should also be annotated with @WebService.
@Webmethod: Annotation denotes that this method will be published and used as a web service. The method must be public as per java doc.
SOAPBinding(style = Style.RPC): defines a SOAP styles web service with binding style as RPC
SOAPBinding(style = Style.DOCUMENT): defines a SOAP styles web service with binding style as DOCUMENT.
QName: is used to create qualified name for HelloWorldServerImplService which maps to the name generated in wsdl.

20. Difference between RPC and Document Styles?

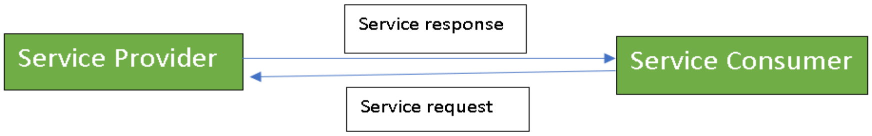
An RPC style web service uses the names of the method and its parameters to generate XML structures representing a method’s call stack. In RPC style, SOAP message is sent as many elements.

Document style indicates the SOAP body contains an XML document which can be validated against pre-defined XML schema document. In document style, SOAP message is sent as a single document.

21. Explain Service Oriented Architecture (SOA) and its uses?

Service-Oriented Architecture (SOA) is an architectural approach in which applications make use of services available in the network. In service oriented architecture, a number of services communicate with each other, in one of two ways: through passing data or through two or more services coordinating an activity.

Service-oriented Architecture:



Service provider: The service provider is the maintainer of the service and the organization that makes available one or more services for others to use. To advertise services, the provider can publish them in a registry, together with a service contract that specifies the nature of the service, how to use it, the requirements for the service, and the fees charged.

Service consumer: The service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service.

22. Explain advantages and disadvantages of SOA?

Advantages:

Service reusability: In SOA, applications are made from existing services. Thus, services can be reused to make many applications.
Easy maintenance: As services are independent of each other they can be updated and modified easily without affecting other services.
Platform independent: SOA allows making a complex application by combining services picked from different sources, independent of the platform.
Availability: SOA facilities are easily available to anyone on request.
Reliability: SOA applications are more reliable because it is easy to debug small services rather than huge codes
Scalability: Services can run on different servers within an environment, this increases scalability

Disadvantages:

High overhead: A validation of input parameters of services is done whenever services interact this decreases performance as it increases load and response time.
High investment: A huge initial investment is required for SOA.
Complex service management: When services interact they exchange messages to tasks. The number of messages may go in millions. It becomes a cumbersome task to handle a large number of messages.

23. Can we have more than one End Points?

Yes we can.

24. What is difference between SOA and Web Services?

Service Oriented Architecture (SOA) is an architectural pattern where applications are designed in terms of services that can be accessed through communication protocol over network. SOA is a design pattern and doesn’t go into implementation.

Web Services can be thought of as Services in SOAP architecture and providing means to implement SOA pattern.

25. What is JAX-WS API?

JAX-WS stands for Java API for XML Web Services. JAX-WS is XML based Java API to build web services server and client application